MMA: Multi-Metric-Autoencoder for Analyzing High-Dimensional and Incomplete Data

Cheng Liang[1], Di Wu[2(✉)], Yi He[3], Teng Huang[1], Zhong Chen[4], and Xin Luo[2]

[1] Institute of Artificial Intelligence and Blockchain, Guangzhou University, Guangzhou 510006, China
c_liang@e.gzhu.edu.cn,huangteng1220@gzhu.edu.cn
[2] College of Computer and Information Science, Southwest University, Chongqing 400715, China
wudi.cigit@gmail.com
[3] Department of Computer Science, Old Dominion University, Norfolk, VA 23529, USA
yihe@cs.odu.edu
[4] Department of Computer Science, Xavier University of Louisiana, New Orleans, LA 70125, USA
zchen@xula.edu

Abstract. High-dimensional and incomplete (HDI) data usually arise in various complex applications, e.g., bioinformatics and recommender systems, making them commonly heterogeneous and inclusive. Deep neural networks (DNNs)-based approaches have provided state-of-the-art representation learning performance on HDI data. However, most prior studies adopt fixed and exclusive L_2-norm-oriented *loss* and *regularization* terms. Such a single-metric-oriented model yields limited performance on heterogeneous and inclusive HDI data. Motivated by this, we propose a Multi-Metric-Autoencoder (MMA) whose main ideas are two-fold: 1) employing different L_p-norms to build four variant Autoencoders, each of which resides in a unique metric representation space with different *loss* and *regularization* terms, and 2) aggregating these Autoencoders with a tailored, self-adaptive weighting strategy. Theoretical analysis guarantees that our MMA could attain a better representation from a set of dispersed metric spaces. Extensive experiments on four real-world datasets demonstrate that our MMA significantly outperforms seven state-of-the-art models. Our code is available at the link https://github.com/wudi1989/MMA/

Keywords: Data Science · High-dimensional and Sparse Matrices · Deep Neural Network-based Representation · Matrix Representation · Multi-Metric Modeling

This work is supported by the National Natural Science Foundation of China under grant 62176070.

D. Koutra et al. (Eds.): ECML PKDD 2023, LNAI 14173, pp. 3–19, 2023.
https://doi.org/10.1007/978-3-031-43424-2_1

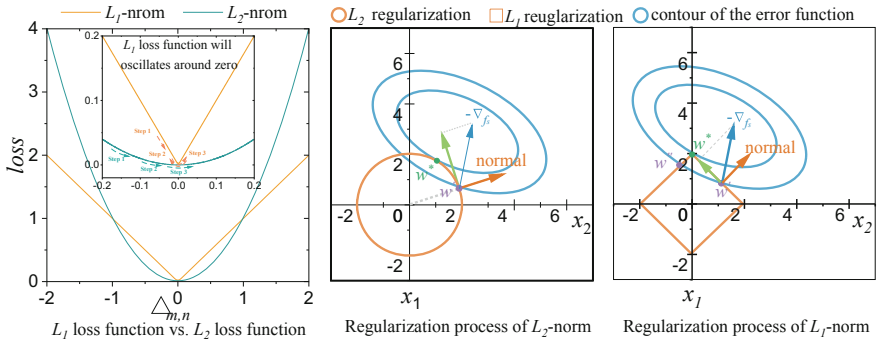


Fig. 1. An example of L_p-norms as *loss* function and *regularization*. The orange/blue/green arrow represents the *regularization*'s normal/gradient/real update direction at the current parameter position w', and w^* is the optimal parameter.

1 Introduction

Matrices are commonly adopted to describe pairwise relationships between entities with a wide range of application scenarios, such as bioinformatics [15], industrial manufacturing [1], and recommendation system (RS) [8,14,32] to mention a few. For instance, the user-item matrix is usually used to record users' behaviors interacting with different items in online shopping systems [35]. Typically, the rows/columns of the user-item matrix represent users/items, and the entries record the results of their interactions.

The core of analyzing and representing these data matrices lies in handling their high-dimensional and incomplete (HDI) characteristics [20,22,47] that conceptually and basically exist in real application scenarios. For example, an RS usually has a large number of items that a user cannot fully access, leaving the user behavior data to be highly incomplete [13,31]. Moreover, the different types of interactions and systems mold the diversity of data matrices. Hence, well-designed methods are desired to represent HDI matrices for excavating hidden patterns and knowledge [3,21,43].

With the rapid development of deep learning, deep neural network (DNNs) has been widely adopted for analyzing HDI matrix [32] and provided state-of-the-art representation performance. Despite their success, they share the same essence: their *loss* and *regularization* terms are exclusive and fixed, i.e., they are a single-metric-oriented model. However, the real-world HDI data are heterogeneous and diversified with various underlying properties [2,33], manifesting the limitations of such single-metric-oriented model [19,42]. A brief example is provided in Fig 1 to discuss the differences of L_p-norms as *loss* function and *regularization*, respectively.

– **Difference between L_p-norms as *loss* function.** Let $\triangle_{j,k} = x_{j,k} - \hat{x}_{j,k}$ denotes the distance between the ground truth and the prediction, L_1-norm-oriented *loss* is $l(\triangle_{j,k}) = \|\triangle_{j,k}\|_{L_1}$, and L_2-norm-oriented *loss* is $l(\triangle_{j,k}) =$

$\|\triangle_{j,k}\|_{L_2}^2$ (left panel of Fig 1). The significant differences between them are three-fold: 1) Robustness. L_2-norm-oriented *loss* is more sensitive to errors, making it less robust than L_1-norm-oriented *loss* to the outliers; 2) Stability. Since L_1-norm-oriented *loss* has vertices causing the gradient of L_1-norm-oriented *loss* oscillates around zero, while L_2-norm-oriented is smooth, so the *loss* is stable; 3) Uniqueness of solution. There are multi-solution of L_1-norm-oriented *loss* in high dimensional space, while L_2-norm-oriented *loss* has a unique solution.

– **Difference between L_p-norms as *regularization*.** Regularization is used to prevent overfitting. Let L_1-norm regularization be $r(w_i) = \|w_i\|_{L_1}$, and L_2-norm regularization be $r(w_i) = \|w_i\|_{L_2}^2$, as shown in the middle and right panels of Fig. 1. Their major difference is their contour shape, i.e., L_1-norm regularization's contour shape is a rectangle, while L_2-norm regularization is a circle. Then, they have two-fold different consequences: 1) Sparse parameters. The updating process of L_1-norm regularization is mostly stopped at its vertices, resulting in the regularized parameters being mostly zero, providing a feature selection effect. 2) Fine representation. The parameters regularized by L_2-norm are mostly non-zero, which can fine-tune the value of parameters for a better representation.

We offer an affirmative answer by proposing a Multi-Metric-Autoencoder (MMA) with a two-fold main idea: 1) employing different L_p-norms to build four variant Autoencoders, each of which resides in a unique metric space with distinct *loss* and *regularization* terms, and 2) aggregating these Autoencoders with a tailored, self-adaptive weighting strategy. By doing so, the proposed MMA enjoys a better multi-metric-based learning ability, achieving non-biased and comprehensive representations of HDI data. This paper has the following contributions:

– It first proposes to exploit a multi-metric strategy to boost a DNNs-based model's representation learning ability on the HDI matrix.
– It proposes a Multi-Metric-Autoencoder (MMA) that can accurately represent an HDI matrix by ensembling four variant Autoencoders built with different L_p-norms.
– Empirical studies verify that the proposed MMA can aggregate the merits from disparate metric spaces with different L_p-norms.
– Algorithm design and comprehensive analyses are provided for the proposed MMA.

Experimental results on four real-world benchmark datasets demonstrate that the proposed MMA significantly outperforms both non-DNN-based and DNN-based single-metric state-of-the-art models.

2 Related Work

2.1 LFA-Based Model

Latent factor analysis-based (LFA-based) models are widely used to represent HDI matrices [10,44]. However, prior works have mainly used only one fixed *loss* function or *regularization* to train their models. Examples include matrix factorization based [37], generalized non-negative with momentum-based [24], linked open data-based [26], dual regularization based [29], mapreduce-based [11], and generalized nesterov-based [23] to name a few. Although the L_2 *loss* function has been shown to be more sensitive to outliers but stable during training [34], the L_1 *loss* function is more robust to outliers but unstable during training [18], and L_1 *regularization* contains the built-in feature selection characterize [18]. Hence, LFA-based representation methods trained under different L_p-norms have been proposed, including elastic-net regression regularize-based [28], and smooth L_1-norm-oriented [40]. However, these studies are built on the linear MF model, and they only discuss different metrics on *loss* function while ignoring the *regularization*. Thus, there is currently a lack of discussion on multi-metric DNN-based HDI matrix representation methods built on different L_p-norms.

2.2 Deep Learning-Based Model

The recent development of deep learning has led to the widespread use of deep learning-based models for representing HDI data, owing to their ability to learn highly non-linear representations [38,41,47]. Comprehensive reviews of recent DNN-based HDI matrix representation methods have been conducted by some researchers [47]. Among these methods, one of the representative methods is autoencoder-based [30], which has inspired the development of variational autoencoder-based methods [6], kernelized synaptic-based autoencoder [25] and global and Local Kernels-based autoencoder [9]. Other notable research includes neural rating regression-based [17], unsupervised disentangle representation [39], unsupervised hyperbolic representation [27], and dual metric learning-based [36]. In addition, GNN-based representation methods [7,12] have also been proposed for representing HDI data, such as inductive matrix completion-based methods [45].

Compared to the aforementioned methods, MMA holds the following significance: 1) DNN-based models possess greater data representation capabilities owing to their nonlinear features compared to LFA-based models; 2) DNN-based models do not require complex graph data in contrast to GNN-based models, resulting in lower computational resource consumption; 3) To the best of our knowledge, MMA is the first work to investigate the performance of multi-metrics-oriented DNN-based models regarding representing HDI data. Section 3 of the paper presents numerical experiments and comparisons with state-of-the-art models, demonstrating the superior performance of the proposed MMA.

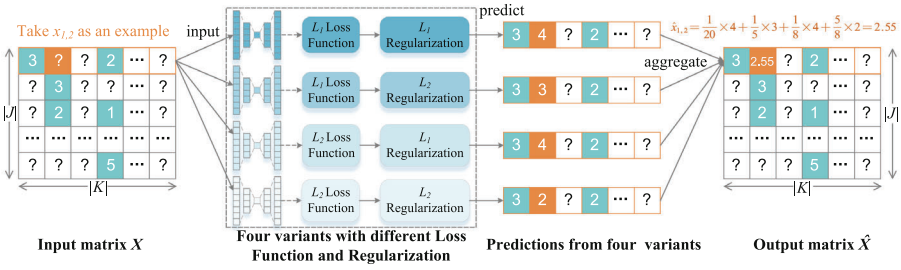


Fig. 2. The architecture and predicting process of the proposed MMA

3 Methodology

To aggregate the advantages of different L_p-norm, we propose the MMA model. Figure 2 demonstrates the architecture and predicting process of MMA, which can be divided into two parts: 1) Utilization of the observed entries of X to train four variant models with independent parameters and predict the unobserved entries; 2) Ensemble these predictions with a tailored self-adaptive weighting strategy to obtain the final output. The establishment of the variant, the self-adaptively weighting strategy, and the theoretical analysis are depicted in the following contents.

3.1 Establishment of Base Model

We choose the representative I-AutoRec [30] to be the basic model in this paper. Formally, given a vector $\mathbf{x}^{(k)} \in X$, I-AutoRec completes the unknown entities by excavating the hidden pattern from known entries. Its error function $\mathcal{L}(\cdot)$ is:

$$\mathcal{L}(f) = \sum_{\mathbf{x}^{(k)} \in X} \left\| \left(\mathbf{x}^{(k)} - f\left(\mathbf{x}^{(k)}; \theta\right) \right) \odot \mathbf{m}^{(k)} \right\|_{L_2}^2 + \frac{\lambda}{2} \cdot \sum_{i \in I} \|w_i\|_{L_2}^2 , \tag{1}$$

where $\lambda > 0$ is the *regularization* factor, $\theta = \{w_i, b_i\}(i \in (1, 2, \cdots, I)$ denotes the weighted term and bias term of i_{th} hidden layer, $\|\cdot\|_{L_2}^2$ is the square of L_2-norm, $\odot$ represents the Hadamard product, and vector $\mathbf{x}^{(k)} = \{x_{1,k}, \cdots, x_{|J|,k}\}$ of k column of X, $\mathbf{m}^{(k)}$ is the k_{th} column of index matrix M for I-AutoRec in which $m_{j,k}$ records whether $x_{j,k}$ is observed, $m_{j,k} = 1$ if there is, $m_{j,k} = 0$ otherwise.

We employ different L_p-norms in *loss* function $l(\cdot)$ and *regularization* $r(\cdot)$ to establish four base variant models summarized in Table 1.

MMA-1 (L_1 *loss* function and L_1 *Regularization*). The objective function for MMA-1 to minimize is as follows:

$$\mathcal{L}^1(f) = \sum_{\mathbf{x}^{(k)} \in X} \left\| \left(\mathbf{x}^{(k)} - f^1\left(\mathbf{x}^{(k)}; \theta\right) \right) \odot \mathbf{m}^{(k)} \right\|_{L_1} + \frac{\lambda}{2} \cdot \sum_{i \in I} \|w_i\|_{L_1}. \tag{2}$$

Table 1. Summarization of four base variant models

base models	$l(\triangle_{j,k})$	$r(w_i)$	Characteristic
MMA-1	$\lvert\triangle_{j,k}\rvert$	$\lVert w_i\rVert_1$	Robustness & Feature Selection
MMA-2	$\lvert\triangle_{j,k}\rvert$	$\lVert w_i\rVert_2^2$	Robustness & Fine representation
MMA-3	$(\triangle_{j,k})^2$	$\lVert w_i\rVert_1$	Stability & Feature Selection
MMA-4	$(\triangle_{j,k})^2$	$\lVert w_i\rVert_2^2$	Stability & Fine representation

MMA-2 (L_1 *loss* function and L_2 *Regularization*). The objective function for MMA-2 to minimize is as follows:

$$\mathcal{L}^2(f) = \sum_{\mathbf{x}^{(k)} \in X} \left\| \left(\mathbf{x}^{(k)} - f^2\left(\mathbf{x}^{(k)}; \theta\right) \right) \odot \mathbf{m}^{(k)} \right\|_{L_1} + \frac{\lambda}{2} \cdot \sum_{i \in I} \|w_i\|_{L_2}^2. \quad (3)$$

MMA-3 (L_2 *loss* function and L_1 *Regularization*). The objective function for MMA-3 to minimize is as follows:

$$\mathcal{L}^3(f) = \sum_{\mathbf{x}^{(k)} \in X} \left\| \left(\mathbf{x}^{(k)} - f^3\left(\mathbf{x}^{(k)}; \theta\right) \right) \odot \mathbf{m}^{(k)} \right\|_{L_2}^2 + \frac{\lambda}{2} \cdot \sum_{i \in I} \|w_1\|_{L_1}. \quad (4)$$

MMA-4 (L_2 *loss* function and L_2 *Regularization*). The objective function for MMA-4 to minimize is as follows:

$$\mathcal{L}^4(f) = \sum_{\mathbf{x}^{(k)} \in X} \left\| \left(\mathbf{x}^{(k)} - f^4\left(\mathbf{x}^{(k)}; \theta\right) \right) \odot \mathbf{m}^{(k)} \right\|_{L_2}^2 + \frac{\lambda}{2} \cdot \sum_{i \in I} \|w_1\|_{L_2}^2. \quad (5)$$

3.2 Self-Adaptively Aggregation

Ensemble learning is an excellent method for aggregating multi-models. It requires the base model to be diversified and accurate. The usage of different L_p-norm guarantees diversity. And the representative I-AutoRec guarantees accuracy. Thus, the base models satisfy the two requirements of the ensemble. We adopt the self-adaptively aggregation method in the validation set according to their *loss*. The principle of this idea is to increase the weight of the t_{th} base model if its *loss* decreases in the n_{th} training iteration or reduce its weight otherwise. We present the related definitions to explain the theoretical analysis better.

Definition 1 (Separate *Loss* of Base Models). We use $Sl^t(n)$ to denote the separate *loss* of the t_{th} base model at n_{th} iteration, which can be computed as follows:

$$Sl^t(n) = \sqrt{\sum_{j \in J, k \in K, (J,K \in \Gamma)} \left(\left(x_{j,k} - \hat{x}_{j,k}^t \right) \times m_{j,k} \right)^2 / \|\Gamma\|_0}$$
$$\hat{x}_{j,k}^t = f^t(j,k;\theta) \quad \text{s.t.} \quad t = 1,2,3,4, \quad (6)$$

where $\|\cdot\|_0$ represents the L_0-norm of a matrix which indicates the number of non-zero elements in it, and Γ denotes the validation subset of X.

Definition 2 (Accumulative *Loss* of Base Models). The accumulative *loss* $Al^t(n)$ of Sl^t until n_{th} training iteration can be computed as follows:

$$Al^t(n) = \sum_{h=1}^{n} Sl^t(h). \tag{7}$$

Definition 3 (Ensemble Weight of Base Models). The ensemble weight ε^t of the t_{th} base model can be calculated as follows:

$$\varepsilon^t(n) = \frac{e^{-\delta Al^t(n)}}{\sum_{t=1}^{4} e^{-\delta Al^t(n)}}, \tag{8}$$

where δ is the equilibrium factor that controls the ensemble aggregation weights during the training process. Based on Definitions 1–3, the final prediction of MMA in n_{th} training iteration can be represented as follows:

$$\hat{x}_{j,k} = \sum_{t=1}^{4} \varepsilon^t(n)\hat{x}^t_{j,k}. \tag{9}$$

3.3 Theoretical Analysis

Definition 4 (Separate *Loss* of MMA). We use $Sl(n)$ to represent the *loss* of MMA at n_{th} training iteration, which is computed as follows:

$$Sl(n) = \sqrt{\sum_{j\in J,k\in K,(J,K\in\Gamma)} \left((x_{j,k} - \hat{x}_{j,k}) \times m_{j,k}\right)^2 / \|\Gamma\|_0} \tag{10}$$

where $\hat{x}_{j,k}$ is calculated by align (9).

Definition 5 (Accumulative *Loss* of MMA). The accumulative *loss* of MMA can be represented as follows:

$$Al(n) = \sum_{h=1}^{n} Sl(h). \tag{11}$$

Theorem 1. *Given an MMA model, supposing the $Al^t(n)$ of base models lies between [0,1]. If $\varepsilon^t(n)$ sets as align (8) during the training, then the following align holds:*

$$Al(N) \leq \min\left\{Al^t(N) \mid t = 1, 2, 3, 4\right\} + \frac{\ln 4}{\delta} + \frac{\delta N}{8}, \tag{12}$$

where N is the maximum iteration.

By setting $\delta = \sqrt{1/\ln N}$ of *Theorem* 1, the upper bound becomes:

$$Al(N) \leq \min\left\{Al^t(N) \,|\, t = 1, 2, 3, 4\right\} + \ln 4\sqrt{\ln N} + \frac{N}{8\sqrt{\ln N}}, \tag{13}$$

where $\ln 4\sqrt{\ln N} + \frac{N}{8\sqrt{\ln N}}$ is bound by N linearly. Then we have the following proposition.

Proposition 1. *Gvien* $\delta = \frac{1}{\sqrt{\ln N}}$ *the inequality holds:*

$$Al(N) \leq \min\left\{Al^t(N) \,|\, t = 1, 2, 3, 4\right\} + const, \tag{14}$$

where $\lim_{N\to\infty} const = 19.45$.

Remark 1. **Proposition** 1 indicates that $Al(N)$ is constrained by $\min\{Al^t(N) \mid t = 1, 2, 3, 4\} + const$ with $\delta = \sqrt{1/\ln N}$. Remarkably, each base variant with different L_p-norms as *loss* function and *regularization* leave them to enjoy separate metric space. The ensemble weight in align (8) guarantees that the MMA's *loss* is always lesser than that of the base models, empowering it to have multi-metric orientation from different L_p-norms. In addition, **Proposition** 1 is not intended to demonstrate the accuracy improvement of MMA on the test set, but rather to establish that the model possesses the advantages of the basic models. By showing that the proposed model achieves a smaller loss compared to each individual basic model used separately, it indicates that the model retains the respective strengths of the basic models without compromising its ability to fit the data.

4 Experiments

In this session, the subsequent experiments are aimed at answering the following research questions (RQs):

- RQ. 1. Does the proposed MMA outperform state-of-the-art models in representing the original and outlier-added HDI matrix?
- RQ. 2. How does the MMA self-adaptively control the ensemble weights of its based models during the training process to guarantee performance?
- RQ. 3. Are the base models of MMA diversified in representing the identical HDI matrix to ensure better performance of MMA?

4.1 General Settings

Datasets. Four commonly used real-world HDI datasets are collected to conduct the subsequent experiments. The details of these datasets are summarized in Table 2. These HDI matrix datasets include MovieLens_1M (D1), MovieLens_100k (D2), MovieLens_HetRec (D3), and Yahoo (D4). We adopt a 70%-10%-20% train-validate-test division ratio for all datasets in all experiments involved.

Table 2. Properties of all the datasets.

No.	Name	$\|\mathbf{M}\|$	$\|\mathbf{N}\|$	$\boldsymbol{H_o}$	Density*
D1	MovieLens_1M	6040	3952	1 000 209	4.19%
D2	MovieLens_100k	943	1682	100 000	6.30%
D3	MovieLens_HetRec	2113	10 109	855 598	4.01%
D4	Yahoo	15 400	1000	365 704	2.37%

* Density denotes the percentage of observed entries in the user-item matrix.

Evaluation Metrics. The primary purpose of representing the HDI matrix is to predict its missing entries. To evaluate the representation's accuracy of the tested model, we adopt the root mean square error (RMSE) and mean absolute error (MAE) as the evaluation metrics according to [40,42]. In addition, we have adopted commonly used metrics in recommendation systems, namely Normalized Discounted Cumulative Gain (NDCG) and Hit Ratio to evaluate the performance of our proposed model in the specific task of ranking prediction. To calculate NDCG and Hit Ratio, We select k items that users like (with ratings greater than or equal to 4) as positive samples and pair them with 100 negative samples (with ratings lower than 4). If a positive sample ranks within the top k in the recommendation list, it is considered a hit. The NDCG calculation formula is $NDCG_{u,i} = 1/(log2(rank_{u,i} + 1))$. By calculating the average metric values of positive samples for each user, we obtain the NDCG and Hit Ratio results.

Baselines. The proposed MMA model is compared with seven state-of-the-art HDI data representation models, including one original model AutoRec [30], two Latent factor analysis-based (LFA-based) models MF [16] and FML [46], and four DNN-based models NRR [17], SparseFC [25], IGMC [45], and GLocal-K [9]. Table 3 briefly describes these competitors.

Implementation Details. The hyper-parameters of all the models involved are fine-tuned to achieve optimal performance. Expressly, we set the regularization coefficient of MMA-1 and MMA-2 to 0.01 and that of MMA-3 and MMA-4 to 30 to get a comprehensive high-performance MMA to all datasets. And the learning rate of all variants is set to 0.001 for a better representation performance. The final testing result is output by the optimal model that reaches the lowest prediction error in the validation set during the training process. The training procedure of a model will terminate if its training iterations reach the preset threshold, and all the experiments are conducted on a GPU server with two 2.4 GHz Xeon Gold 6240R with 24 cores, 376.40 GB RAM, and 4 T V100 GPU.

4.2 Performance Comparison (RQ.1)

Comparison of Missing Entries Prediction. Table 4 records the prediction accuracy of all models involved in D1 to D4. The statistical analysis of the loss/tie/win, the Wilcoxon signed-ranks test [4], and the Friedman test [4] are

Table 3. Descriptions of all the contrasting models.

Model	Description
MF [16]	The representative LFA-based model factorizes the HDI matrix for data representation. *Computer 2009*.
AutoRec [30]	The representative autoencoder-based model encodes HDI data into low-dimension representation and decodes them to finish the prediction. *WWW 2015*.
NRR [17]	It is a DNN-based multi-task learning framework for HDI data representation. *SIGIR 2017*.
SparceFC [25]	A DNN-based model parametrizes the weighted matrices into low-dimensional vectors to capture essential features of the HDI matrices. *ICML 2018*.
IGMC [45]	It is a GNN-based model which can excavate hidden patterns from graphs built from a HDI matrix without using side information. *ICLR 2020*.
FML [46]	An LFA-based model combines metric learning (distance space) with collaborative filtering. *IEEE TII 2020*.
GLocal-K [9]	A DNN-based model generalizes and represents user behavior data into a low-dimensional space with the fine-tuned global kernel. *CIKM 2021*

Table 4. The comparison of the prediction accuracy of MMA and its competitors, including the loss/tie/win counts, Wilcoxon signed-ranks test, and Friedman test.

Dataset	Metric	MF	AutoRec	NRR	SparseFC	IGMC	FML	Glocal-K	MMA **(ours)**
D1	RMSE	0.857•	0.847•	0.881•	0.839○	0.867•	0.849•	0.839○	0.840
	MAE	0.673•	0.667•	0.691•	0.656	0.681•	0.667•	0.655○	0.656
D2	RMSE	0.913•	0.897•	0.923•	0.899•	0.915•	0.904•	0.892•	0.889
	MAE	0.719•	0.706•	0.725•	0.706•	0.722•	0.718•	0.697•	0.695
D3	RMSE	0.757•	0.752•	0.774•	0.749•	0.769•	0.754•	0.756•	0.744
	MAE	0.572•	0.569•	0.583•	0.567•	0.582•	0.573•	0.573•	0.561
D4	RMSE	1.206•	1.172•	1.227•	1.203•	1.133○	1.176•	1.204•	1.163
	MAE	0.937•	0.900•	0.949•	0.915•	0.848○	0.937•	0.905•	0.879
Statistic	loss/tie/win	0/0/8	0/0/8	0/0/8	1/1/6	2/0/6	0/0/8	2/0/6	**5/1/50***
	p-value	0.006	0.006	0.005	0.014	0.018	0.006	0.018	-
	F-rank	5.938	3.375	8.0	3.188	5.5	4.938	3.375	**1.688**

*The total loss/tie/win cases of MMA. • The cases that MMA wins the other models in comparison. ○ The cases that MMA loses the comparison.

made to analyze these results better. These results are presented in the third-to-last, second-to-last, and last row of Table 4. From Table 4, we observe that MMA achieves the lowest RMSE/MAE in most cases, and the total loss/tie/win cases are 5/1/50. Its p-value is lower than the significance level of 0.05 and achieves the lowest F-rank among all the participants. These observations reveal that MMA achieves a better representation accuracy for the HDI matrix than other models.

Comparison of Ranking Prediction. We have also conducted statistical analysis on the results of NDCG and Hit Ratio and recorded them in Table 5.

Table 5. The NDCG and Hit Ratio of MMA and its competitors.

Dataset	Metric	MF	AutoRec	NRR	SparseFC	IGMC	FML	Glocal-K	MMA (ours)
D1	NDCG@5	0.550•	0.551•	0.520•	0.565○	0.532•	0.563•	0.567○	0.564
	Hit@5	0.746•	0.749•	0.717•	0.758	0.728•	0.755•	0.758	0.758
	NDCG@10	0.599•	0.602•	0.568•	0.608	0.571•	0.606•	0.608	0.608
	Hit@10	0.883•	0.887•	0.866•	0.892○	0.875•	0.887•	0.891○	0.890
D2	NDCG@5	0.529•	0.542•	0.509•	0.541•	0.519•	0.541•	0.548•	0.559
	Hit@5	0.738•	0.742•	0.717•	0.750•	0.730•	0.744•	0.761•	0.761
	NDCG@10	0.579•	0.593•	0.564•	0.592•	0.568•	0.592•	0.595•	0.608
	Hit@10	0.892•	0.897•	0.886•	0.904•	0.889•	0.899•	0.904•	0.911
D3	NDCG@5	0.479•	0.483•	0.455•	0.491•	0.461•	0.486•	0.473•	0.493
	Hit@5	0.645•	0.648•	0.623•	0.660•	0.631•	0.651•	0.644•	0.662
	NDCG@10	0.526•	0.529•	0.504•	0.540•	0.511•	0.532•	0.520•	0.542
	Hit@10	0.789•	0.791•	0.773•	0.800•	0.779•	0.794•	0.790•	0.802
D4	NDCG@5	0.526•	0.574•	0.520•	0.558•	0.612○	0.580•	0.552•	0.603
	Hit@5	0.789•	0.802•	0.764•	0.794•	0.835○	0.807•	0.795•	0.828
	NDCG@10	0.549•	0.628•	0.578•	0.612•	0.659○	0.631•	0.608•	0.651
	Hit@10	0.784•	0.960•	0.942•	0.958•	0.979○	0.961•	0.966•	0.973
Statistic	loss/tie/win	0/0/16	0/0/16	0/0/16	2/2/12	4/0/12	0/0/16	2/2/12	**8/4/100***
	p-value	0.002	0.0002	0.0002	0.0014	0.0013	0.0278	0.0015	-
	F-rank	5.938	3.969	7.563	3.094	5.438	4.844	3.375	**1.781**

* The total loss/tie/win cases of MMA. • The cases that MMA wins the other models in comparison. ○ The cases that MMA loses the comparison.

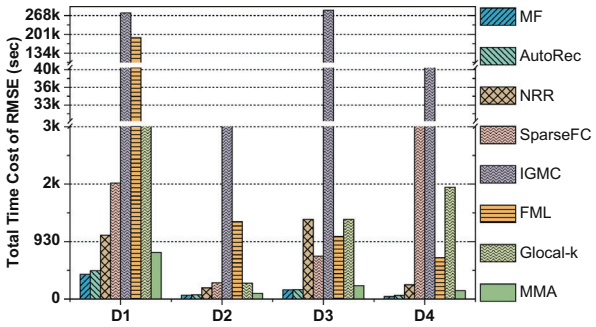
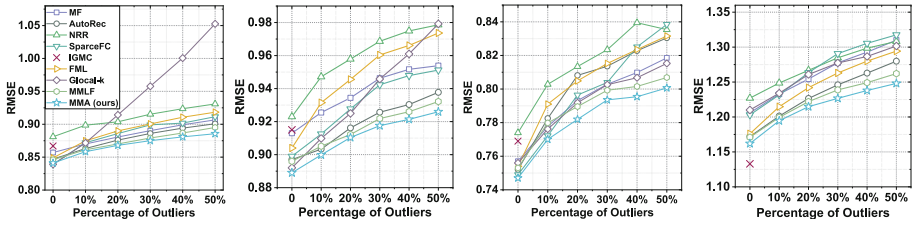


Fig. 3. The histogram graph of the total time cost to reach the optimal accuracy of all the participating models.

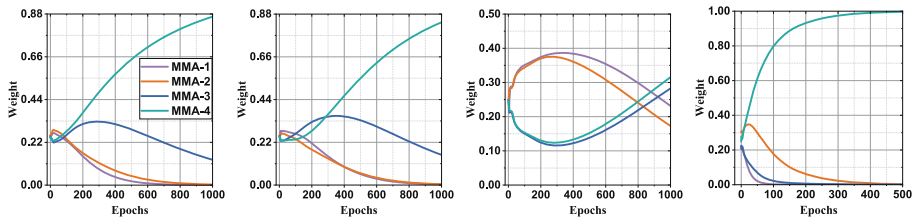
From the results in Table 5, we can observe that our proposed model outperforms other comparison methods in most cases, with a final score of 8/4/100 in terms of loss/tie/win. Moreover, the p-values are below the significant value of 0.05, and MMA also reaches the lowest F-rank.

Comparison of Computational Efficiency. Figure 3 displays the total time all models require to reach the optimal RMSE on the validation set. As shown in Fig. 3, LFA-based models are more efficient than DNN-based models as they only train on observed data. Additionally, due to its complex data form and architecture, the GNN-based model consumes significant computational resources and time. Our proposed MMA is with acceptable time consumption, which lands between the LFA-based and GNN-based models. Compared to the original model, MMA's growth time mainly focuses on calculating ensemble weights since its basic model is trained in parallel. The time ratios for D1 to D4 growth



(a) D1 (b) D2 (c) D3 (d) D4

Fig. 4. The robustness testing results under the RMSE metric of MMA and its competitors.



(a) D1 (b) D2 (c) D3 (d) D4

Fig. 5. The curve graph of the changes of ensemble weights during the training process.

are: 64%, 37.5%, 43.2%, and 43.2%. As a result, we conclude that MMA can outperform other state-of-the-art models with only minor sacrifices in terms of computational resources and time.

Robustness of MMA to Outlier Data. As previously discussed, the L_1 *loss* function has been shown to be more robust to outlier data. To investigate if MMA can inherit this robustness from the ensemble, we compare its accuracy on datasets with 10% to 50% outliers added, which includes two steps: 1) Select 10% to 50% of the known entries from the original dataset, and 2) randomly assign maximum or minimum known value to these entries. In reality, HDI matrices often contain various outliers that can impair the accuracy of the representation, making robustness an important evaluation indicator for HDI data representation.

Figure 4 records the accuracy of different models' RMSE on D1 to D4 and Fig. IGMC's computational efficiency is limited due to its time-consuming graph construction and operations, as evidenced by the over 70 h running the original data set (Fig. 3). With increasing noise and data set size, its performance further deteriorates. We omit its results after a five-day failed run. Figure 4 and Fig. S2 show that MMA outperforms its competitors in all cases, e.g., on D1 with 50% outliers, MMA's RMSE/MAE is 0.886/0.691, while most other models are above 0.9/0.7, indicating that it benefits from different metrics to obtain robustness.

Table 6. The comparison of the prediction accuracy of MMA and its base models with statistical analysis.

Dataset	Metric	MMA-1	MMA-2	MMA-3	MMA-4	MMA
D1	RMSE	0.867●	0.859●	0.849●	0.847●	0.840
	MAE	0.663●	0.660●	0.668●	0.667●	0.656
D2	RMSE	0.909●	0.906●	0.898●	0.897●	0.889
	MAE	0.700●	0.698●	0.706●	0.706●	0.695
D3	RMSE	0.755●	0.755●	0.752●	0.751●	0.744
	MAE	0.563●	0.563●	0.571●	0.569●	0.561
D4	RMSE	1.198●	1.194●	1.185●	1.173●	1.161
	MAE	0.870○	0.887●	0.900●	0.900●	0.879
Statistic	loss/tie/win	1/0/7	0/0/8	0/0/8	0/0/8	**1/0/32***
	p-value	0.025	0.006	0.006	0.006	-
	F-rank	3.625	3.250	3.875	3.125	1.125

* The MMA's total loss/tie/win cases. ● The cases that MMA wins the base models in comparison. ○ The cases that MMA loses the comparison.

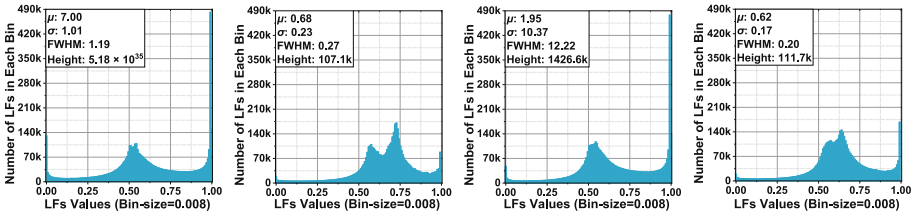
4.3 The Self-ensembling of MMA (RQ.2)

We further record the ensemble weights during the training process and compare the accuracy between MMA and its base models to discuss how MMA achieves self-adaptively control empirically.

Monitoring the Variations of Ensemble Weights. Figure5 demonstrates the ensemble weight variations of D1 to D4. In Fig 5, we find that all the weights self-adaptively change across the training process. We note that the ensemble weights of MMA-3 and MMA-4, calculated according to Equation (8) on the validation set, gradually increase and surpass MMA-1 and MMA-2 until the base models are fitted. The reason may be that this dataset has fewer outliers, making the L_2-norm *loss* function dominant during the training process to achieve better predictions.

Accuracy Comparison between MMA and its Base Models. The prediction accuracy of MMA and the base variant models are recorded in Table 6. In like manner, we also conduct the loss/tie/win count, the Wilcoxon singed-rank test, and the Friedman test. Table 6 demonstrates that the MAE of MMA-1 and MMA-2 are generally lower than MMA-3 and MMA-4, but RMSE is the opposite.

Summary. We can conclude that MMA does take advantage of different metric spaces. MMA outperforms other state-of-the-art models with minor computational resource sacrifice by ensembling different metric spaces in the aggregation stage.



(a) MMA-1 (b) MMA-2 (c) MMA-3 (d) MMA-4

Fig. 6. The distribution histogram of latent factors (LFs) of the base models on D3.

4.4 Base Models' Latent Factors Distribution (RQ. 3)

To verify if the base models of MMA are diversified in representing the HDI matrix, we visualize the encoder output of these base models, which we regard as its latent factors. We adopt the Gaussian Function for better analysis. Figure 6 depicts the LF's distribution of the base models on D3. The figure shows that MMA-1 and MMA-3 distribute around 0 and 1 because parameters regularized by L_1-norm are sparse due to their feature selection characteristic.

These observations prove the base models are diversified in representing the HDI matrix. According to the principle of the ensemble theory [5], such variation is beneficial for boosting a base model, guaranteeing the representation accuracy of MMA on HDI data.

5 Conclusion

This paper proposes a multi-metric-Autoencoder to represent HDI data matrices more accurately. Its essential idea is two-fold: 1) deploy different L_p-norms as *loss* function and *regularization* to develop four variant Autoencoders with unique metric spaces, and 2) aggregate these Autoencoders with self-adaptively weighting ensemble strategy. Extensive theoretical and empirical studies verify that the proposed MMA model is benefited from the multi-metrics of basic models. Experiments on four real-world datasets prove that 1) MMA has remarkably higher accuracy and better robustness in representing HDI data, and 2) its computational efficiency is higher than most DNN-based models. In addition, the loss functions of the four different basic models can also be integrated into a single loss function and applied to other types of models, such as latent factor analysis models. We hope to expand the work of MMA in these aspects in the future.

References

1. Alhayani, B.S., et al.: Visual sensor intelligent module based image transmission in industrial manufacturing for monitoring and manipulation problems. J. Intell. Manuf. **32**(2), 597–610 (2021)
2. Cai, D., Qian, S., Fang, Q., Xu, C.: Heterogeneous hierarchical feature aggregation network for personalized micro-video recommendation. IEEE Trans. Multimedia **24**, 805–818 (2021)
3. Chen, J., Luo, X., Zhou, M.: Hierarchical particle swarm optimization-incorporated latent factor analysis for large-scale incomplete matrices. IEEE Trans. Big Data **8**(6), 1524–1536 (2021)
4. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. **7**, 1–30 (2006)
5. Dong, X., Yu, Z., Cao, W., Shi, Y., Ma, Q.: A survey on ensemble learning. Front. Comp. Sci. **14**(2), 241–258 (2020)
6. Fang, L., Du, B., Wu, C.: Differentially private recommender system with variational autoencoders. Knowl.-Based Syst. **250**, 109044 (2022)
7. Gao, C., et al.: A survey of graph neural networks for recommender systems: Challenges, methods, and directions. ACM Trans. Recommender Syst. (2023). https://doi.org/10.1145/3568022
8. Gharahighehi, A., Vens, C., Pliakos, K.: Multi-stakeholder news recommendation using hypergraph learning. In: Koprinska, I., et al. (eds.) ECML PKDD 2020. CCIS, vol. 1323, pp. 531–535. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-65965-3_36
9. Han, S.C., Lim, T., Long, S., Burgstaller, B., Poon, J.: Glocal-k: global and local kernels for recommender systems. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp. 3063–3067 (2021)
10. Hu, L., Pan, X., Tang, Z., Luo, X.: A fast fuzzy clustering algorithm for complex networks via a generalized momentum method. IEEE Trans. Fuzzy Syst. **30**(9), 3473–3485 (2021)
11. Hu, L., Yang, S., Luo, X., Yuan, H., Sedraoui, K., Zhou, M.: A distributed framework for large-scale protein-protein interaction data analysis and prediction using mapreduce. IEEE/CAA J. Automatica Sinica **9**(1), 160–172 (2021)
12. Hu, L., Yang, S., Luo, X., Zhou, M.: An algorithm of inductively identifying clusters from attributed graphs. IEEE Trans. Big Data **8**(2), 523–534 (2020)
13. Hu, L., Zhang, J., Pan, X., Luo, X., Yuan, H.: An effective link-based clustering algorithm for detecting overlapping protein complexes in protein-protein interaction networks. IEEE Trans. Netw. Sci. Eng. **8**(4), 3275–3289 (2021)
14. Islek, I., Oguducu, S.G.: A hybrid recommendation system based on bidirectional encoder representations. In: Koprinska, I., et al. (eds.) ECML PKDD 2020. CCIS, vol. 1323, pp. 225–236. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-65965-3_14
15. Khan, S., Huh, J., Ye, J.C.: Adaptive and compressive beamforming using deep learning for medical ultrasound. IEEE Trans. Ultrason. Ferroelectr. Freq. Control **67**(8), 1558–1572 (2020). https://doi.org/10.1109/TUFFC.2020.2977202
16. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
17. Li, P., Wang, Z., Ren, Z., Bing, L., Lam, W.: Neural rating regression with abstractive tips generation for recommendation. In: Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval, pp. 345–354 (2017)

18. Li, Y., Sun, H., Yan, W., Cui, Q.: R-CTSVM+: robust capped L1-norm twin support vector machine with privileged information. Inf. Sci. **574**, 12–32 (2021)
19. Li, Z., Li, S., Bamasag, O.O., Alhothali, A., Luo, X.: Diversified regularization enhanced training for effective manipulator calibration. IEEE Transactions on Neural Networks and Learning Systems, pp. 1–13 (2022). https://doi.org/10.1109/TNNLS.2022.3153039
20. Liu, Z., Luo, X., Wang, Z.: Convergence analysis of single latent factor-dependent, nonnegative, and multiplicative update-based nonnegative latent factor models. IEEE Trans. Neural Netw. Learn. Syst. **32**(4), 1737–1749 (2020)
21. Luo, X., Chen, M., Wu, H., Liu, Z., Yuan, H., Zhou, M.: Adjusting learning depth in nonnegative latent factorization of tensors for accurately modeling temporal patterns in dynamic QoS data. IEEE Trans. Autom. Sci. Eng. **18**(4), 2142–2155 (2021)
22. Luo, X., Wu, H., Li, Z.: NeuLFT: a novel approach to nonlinear canonical polyadic decomposition on high-dimensional incomplete tensors. IEEE Trans. Knowl. Data Eng. **35**(6), 6148–6166 (2023)
23. Luo, X., Zhou, Y., Liu, Z., Hu, L., Zhou, M.: Generalized Nesterov's acceleration-incorporated, non-negative and adaptive latent factor analysis. IEEE Trans. Serv. Comput. **15**(5), 2809–2823 (2021)
24. Luo, X., Zhou, Y., Liu, Z., Zhou, M.: Fast and accurate non-negative latent factor analysis on high-dimensional and sparse matrices in recommender systems. IEEE Trans. Know. Data Eng. 35, 3897–3911 (2021). https://doi.org/10.1109/TKDE.2021.3125252
25. Muller, L., Martel, J., Indiveri, G.: Kernelized synaptic weight matrices. In: International Conference on Machine Learning, pp. 3654–3663. PMLR (2018)
26. Natarajan, S., Vairavasundaram, S., Natarajan, S., Gandomi, A.H.: Resolving data sparsity and cold start problem in collaborative filtering recommender system using linked open data. Expert Syst. Appl. **149**, 113248 (2020)
27. Park, J., Cho, J., Chang, H.J., Choi, J.Y.: Unsupervised hyperbolic representation learning via message passing auto-encoders. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5516–5526 (2021)
28. Raza, S., Ding, C.: A regularized model to trade-off between accuracy and diversity in a news recommender system. In: 2020 IEEE International Conference on Big Data (Big Data), pp. 551–560. IEEE (2020)
29. Saberi-Movahed, F., et al.: Dual regularized unsupervised feature selection based on matrix factorization and minimum redundancy with application in gene selection. Knowl.-Based Syst. **256**, 109884 (2022)
30. Sedhain, S., Menon, A.K., Sanner, S., Xie, L.: AutoRec: autoencoders meet collaborative filtering. In: Proceedings of the 24th International Conference on World Wide Web, pp. 111–112 (2015)
31. Shang, M., Yuan, Y., Luo, X., Zhou, M.: An α-β-divergence-generalized recommender for highly accurate predictions of missing user preferences. IEEE Trans. Cybern. **52**(8), 8006–8018 (2021)
32. Shao, B., Li, X., Bian, G.: A survey of research hotspots and frontier trends of recommendation systems from the perspective of knowledge graph. Expert Syst. Appl. **165**, 113764 (2021)
33. Shi, Q., Liu, M., Li, S., Liu, X., Wang, F., Zhang, L.: A deeply supervised attention metric-based network and an open aerial image dataset for remote sensing change detection. IEEE Trans. Geosci. Remote Sens. **60**, 1–16 (2022). https://doi.org/10.1109/TGRS.2021.3085870

34. Shi, X., Kang, Q., An, J., Zhou, M.: Novel L1 regularized extreme learning machine for soft-sensing of an industrial process. IEEE Trans. Ind. Inf. **18**(2), 1009–1017 (2021)
35. Song, Y., Zhu, Z., Li, M., Yang, G., Luo, X.: Non-negative latent factor analysis-incorporated and feature-weighted fuzzy double c-means clustering for incomplete data. IEEE Trans. Fuzzy Syst. **30**(10), 4165–4176 (2022)
36. Tay, Y., Anh Tuan, L., Hui, S.C.: Latent relational metric learning via memory-based attention for collaborative ranking. In: Proceedings of the 2018 World Wide Web Conference, pp. 729–739 (2018)
37. Wang, H., Hong, Z., Hong, M.: Research on product recommendation based on matrix factorization models fusing user reviews. Appl. Soft Comput. (2022). https://doi.org/10.1016/j.asoc.2022.108971
38. Wang, S., Cao, J., Yu, P.S.: Deep learning for spatio-temporal data mining: a survey. IEEE Trans. Knowl. Data Eng. **34**(8), 3681–3700 (2022). https://doi.org/10.1109/TKDE.2020.3025580
39. Wang, X., Chen, H., Zhou, Y., Ma, J., Zhu, W.: Disentangled representation learning for recommendation. IEEE Trans. Pattern Anal. Mach. Intell. **45**(1), 408–424 (2022)
40. Wu, D., Luo, X.: Robust latent factor analysis for precise representation of high-dimensional and sparse data. IEEE/CAA J. Automatica Sinica **8**(4), 796–805 (2021)
41. Wu, D., Luo, X., He, Y., Zhou, M.: A prediction-sampling-based multilayer-structured latent factor model for accurate representation to high-dimensional and sparse data. IEEE Trans. Neural Netw. Learn. Syst. 1–14 (2022). https://doi.org/10.1109/TNNLS.2022.3200009
42. Wu, D., Zhang, P., He, Y., Luo, X.: A double-space and double-norm ensembled latent factor model for highly accurate web service QoS prediction. IEEE Trans. Serv. Comput. (2022). https://doi.org/10.1109/TSC.2022.3178543
43. Wu, H., Luo, X., Zhou, M., Rawa, M.J., Sedraoui, K., Albeshri, A.: A PID-incorporated latent factorization of tensors approach to dynamically weighted directed network analysis. IEEE/CAA J. Automatica Sinica **9**(3), 533–546 (2021)
44. Yuan, Y., He, Q., Luo, X., Shang, M.: A multilayered-and-randomized latent factor model for high-dimensional and sparse matrices. IEEE Trans. Big Data **8**(3), 784–794 (2020)
45. Zhang, M., Chen, Y.: Inductive matrix completion based on graph neural networks. In: International Conference on Learning Representations (2020). https://openreview.net/forum?id=ByxxgCEYDS
46. Zhang, S., Yao, L., Wu, B., Xu, X., Zhang, X., Zhu, L.: Unraveling metric vector spaces with factorization for recommendation. IEEE Trans. Ind. Inf. **16**(2), 732–742 (2019)
47. Zheng, Y., Wang, D.X.: A survey of recommender systems with multi-objective optimization. Neurocomputing **474**, 141–153 (2022)